Honey, I Shrunk the Image: Image Compression With Deep Convolutional Neural Networks

CS 230: Deep Learning

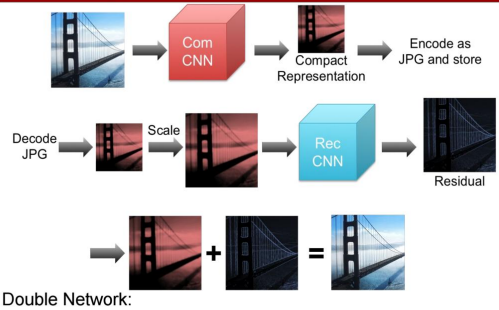
Neel Yerneni, Varun Nambikrishnan, Kiko Ilagan

Stanford Computer Science

Task and Goals

- **Background**: Camera quality continues to improve while image compression techniques lag behind
  - Many image standards lead to noticeable compression artifacts (blocks, blurring, color loss)
  - Image compression using deep learning has become a popular topic in the computer vision research community
- **Task**: Create a system that can compress images that look similar to the original, uncompressed image
- **Goal**: Produce a compressed image with a higher Multi-Scale Structural Similarity (MSSSIM) Index with the original image than JPG, using similar storage space

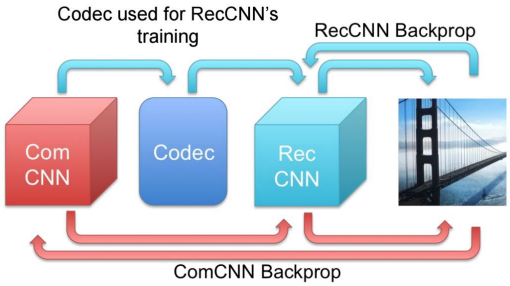
Approach/Architecture



Double Network:

- Train a Compact representation CNN (ComCNN) to learn a compact representation of the image, then compress using an image codec (we use JPG)
- Decode the image and scale it back up to original size using Bicubic Interpolation
- Train a Reconstruction CNN (RecCNN) to learn the residual between the decoded image and the original
- Add the output residual to the decoded image to retrieve the original image

Data and Training



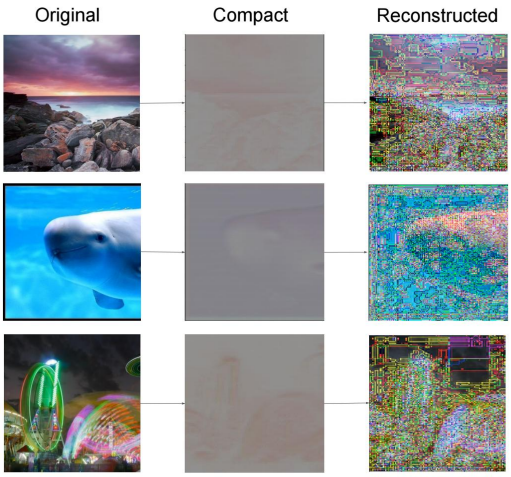
- ComCNN: Loss is the MSE between the original image and the output of RecCNN
- RecCNN: Loss is the MSE between its output and the residual between the original photo and the encoded/decoded/scaled output of ComCNN
- Challenge: codec uses a function that is not differentiable, so one continuous backpropagation is impossible
- Solution: Alternating optimization approach, where the two networks are trained in turns (like a GAN)
  - ComCNN trained under the assumption that the codec perfectly encodes/decodes the compact representation

Results and Analysis

Average MSSSIM by Cohort

	Train	Dev	Test
Network	.8895	.4948	.4628
JPG	-	-	.82

Results and Analysis (cont.)



- Compact representation readily learned by ComCNN and easily compressed with codec - space performance goal met
  - However, heavily polychromatic images proved more difficult to learn
- Reconstruction less successful:
  - Overfitting issues on train set
  - Learned general coloration, but struggled with detail
- Training time takes ~4 days to reach convergence
  - 2 for ComCNN, 2 for RecCNN
  - Greater computational power required for better cross-validation
- Approach shows promise, but will likely need to experiment with altering the loss and more aggressive regularization